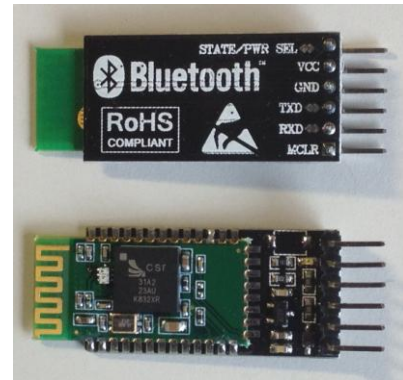


藍牙通訊模組(Slave)

此藍牙模組使用 UART(Tx/Rx)介面來收發訊息, 並能與藍牙手機或平板無線通訊, 其特性如下 :

- 藍牙協定: Bluetooth V2.1 + EDR
- 工作頻率: 2.4~2.48GHz, ISM Band
- 介面: UART
- 輸入電壓: 5V
- 工作溫度: -20°C ~ +75°C



藍牙模組 baud rate : 1200 ~ 115,200 bps
預設值 :

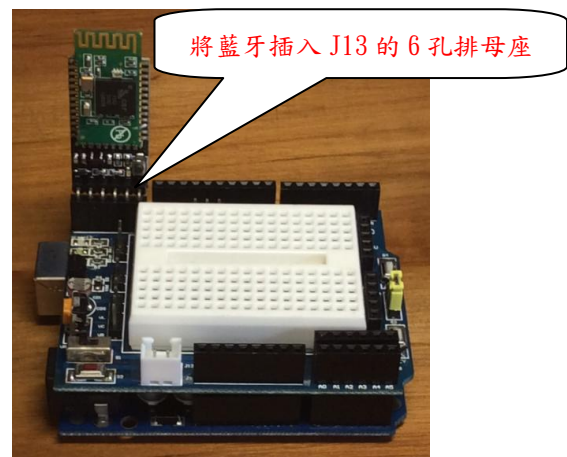
Baud rate : 57,600bps

ID : Appsduino

PIN Code(配對密碼) : 0000

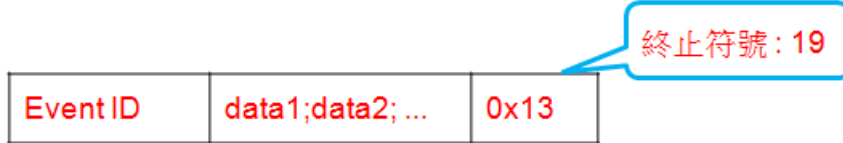
其腳位定義如右 :

Pin	ID	說明
1	PWR SEL	X(沒接)
2	VCC	5V
3	GND	GND
4	TXD	UART TXD(接至 UNO RX Pin/D0)
5	RXD	UART RXD(接至 UNO TX Pin/D1)
6	MCLR	X(重置原先之設定值, 正常沒接)



MeetAndroid Library 命令格式說明：

- Android to Arduino Data Format：



EventID : A ~ Z or a ~ z

起始符號

ex: A1;-0.9876;1.7654 0x13
c1 0x13

MeetAndroid Library 常用函數使用說明：

- AppsduinoBT：使用前需宣告一個 MeetAndroid 的物件，例如：
`MeetAndroid AppsduinoBT ;` //宣告一個 MeetAndroid 的物件名稱叫 AppsduinoBT
- `void registerFunction(void (*)(uint8_t, uint8_t), uint8_t)`：宣告接收到起始符號時的處理程序(processHandler)，例如：
`AppsduinoBT.registerFunction(BTCommHandler, 'c');` //當接收到起始符號為'c'時跳至 BTCommHandler() 函數處理(Call Back Function)
- `bool receive(void)`：檢查是否 received buffer 有接收到資料，若有資料(回傳 true)，並會自動跳至預設的處理程序(processHandler)執行，此函數需置於 loop()內
- `void getString(char[])`：若收到的資料是字串形式，則可以此函數來取出
- `int getInt()`：若收到的資料是整數，則可以此函數來取出
- `long getLong()`：若收到的資料是長整數，則可以此函數來取出
- `float getFloat()`：若收到的資料是浮點數，則可以此函數來取出
- `void send(char)`：可用 send()函數來傳送字元、字串、整數、長整數,... 等不同資料
- `void send(const char[])`：
- `void send(uint8_t)`：
- `void send(int)`：
- `void send(unsigned int)`：
- `void send(long)`：

藍牙接收程式範例如下：

```
#include <MeetAndroid.h> // include Android Bluetooth Library header
MeetAndroid AppsduinoBT; //-- declare one MeetAndroid Bluetooth object --

void setup() {
  Serial.begin(57600); // setup baud rate
  AppsduinoBT.registerFunction(BTCommHandler, 'c'); //register one call back
function
}

void loop() {
  AppsduinoBT.receive() ; // you need to keep this in your loop() to receive events
}

void BTCommHandler (byte flag, byte numOfValues) { //'c' flag to receive BT cmd
  int cmd = AppsduinoBT.getInt();
  If (cmd == ??) do_action ;
  .....
}
```

修改藍牙 ID 名稱的範例程式：(Config_BT_Name.ino)

//若有多個藍牙同時運作，可使用如下(內附)的程式改變 Bluetooth ID 以易於分辨

```
/* ----- BT module configuration program created from 01/24/2013 -----
1. Plug in BT module into 6 Pins BT socket then Power ON
2. Setting BT ID to your specified Name
3. While Green LED turn on means configuration is successful
4. Turn Off Power then take BT module out
5. Repeat step 1 if more BT modules will be configured
*/

int LedGreen = 13 ;

void setup()
{
  pinMode(LedGreen, OUTPUT) ;
  digitalWrite(LedGreen, LOW) ; // turn off LED
  Serial.begin(57600); // Baud rate of HC-0x bluetooth module(設定已知之
速率)
  delay(1000);
  Serial.write("AT+NAMEAppsduino"); // Change the "Appsduino" name to your ID name
  while(!Serial.available()) ; // waiting for 'ok' message
  Serial.read(); // flush feedback message
  digitalWrite(LedGreen, HIGH) ; // turn on LED
}

void loop()
{
}
```

改成你欲設
定之名稱

修改藍牙模組名稱

傳送：AT+NAME`name`

回傳：OK`name`

`name`：所要設置的名稱，需 20 個字元以內。

例：傳送AT+NAMEDino911

回傳：OK`name`

這時藍芽名稱改為Dino911

藍芽模組 Baud rate 修改及速率對應表

指令：AT+BAUD`<p>`

回傳：OK`<r>`

例：要設定 Bluetooth 成 9600 bps，則需送 "AT+BAUD4" command to Bluetooth module.
若成功則回送"OK9600" 訊息！

Note 1：設定時需注意雙方的 Baud Rate 需一致

Note 2：此 BT 模組雖支援至 1,382,400bps(黃色部分)，但 Arduino UNO 控制板之 UART 僅支援至 115,200bps

Note 3：預設之資料格式為 8 data bits, no parity, one stop bit

<code><p></code>	<code><r></code>	備註
1	1200	設定為 1200 bps
2	2400	設定為 2400 bps
3	4800	設定為 4800 bps
4	9600	設定為 9600 bps
5	19200	設定為 19200 bps
6	38400	設定為 38400 bps
7	57600	設定為 57600 bps
8	115200	設定為 115200 bps
9	230400	設定為 230400 bps
A	460800	設定為 460800 bps
B	921600	設定為 921600 bps
C	1382400	設定為 1382400 bps